PGS user manual
version 0.2-0

Kiên Kiêu and Marianne Mora

June 20, 2009

Contents

1 Introduction **2**

2 Installation of pgs package **4**
2.1 Prerequisites 4
2.2 Obtaining and installing pgs 4

3 Lattice of figures **6**
3.1 Classes of geometric objects 6
3.2 Specific generators of some particular planar lattices of figures 6
3.3 Specific generators of some particular 3D lattices of figures . 8
3.4 Generator FigLat 8

4 Planar area prediction **10**
4.1 MSE approximation formulae in the isotropic case 10
4.2 Practical design of a sampling scheme 12
4.3 A posteriori MSE computation 13
4.4 MSE estimation 14
4.4.1 MSE estimation in the isotropic case 14
4.4.2 MSE estimation under mild anisotropy 15

Chapter 1

Introduction

Using stereological methods it is possible to estimate geometric parameters (planar area, volume, surface area, number etc...) of spatial structures (2D or 3D) from partial observations. Partial observations consist of intersections of the spatial structure under study and figures (test sets) such as points, quadrats, lines, planes... In practice, the figures are often systematically distributed in the containing space: the whole sampling device forms a lattice of figures. Such sampling devices are commonly used both because they are easy to implement and because of their statistical efficiency, see e.g. [12].

Provided that the lattice of figures is uniformly randomly translated, most stereological estimators turn out to be unbiased. However assessing the precision of stereological estimators is usually not straightfoward. A naive approach consists of considering the sampling figures as non-correlated statistical units. However as shown by Gundersen & Jensen [12], this approach tends to overestimate the estimation variability. An alternative approach is based on a theory developped by Kendall [15, 16] and Matheron [20, 21]. This approach provides asymptotic approximations of mean square errors (MSE). The approximations converge when the sampling density grows. Stereological applications of Kendall-Matheron theory have been developped in several papers [12, 2, 13, 18, 19, 7, 8, 11, 4, 6, 9, 10, 3, 1, 5, 14].

General MSE formulae for planar area estimation/prediction have been derived [17, 22]. These formulae can be used for a large set of sampling devices : point lattices, point pattern lattices, quadrat lattices, series of parallel lines, strips... Due to their very simple form, the formulae can be used to compare sampling devices independently of the spatial structure under study. Hence they can be very useful in *a priori* sampling design. The R package *pgs* provides tools for computing MSE approximations for area predictors. Since the approximation formulae are easily extended to arbitrary dimensional spaces, MSE approximations for (3D) volume predictors are also provided.

Note that the current version (0.2-0) of the package is still at an experi-

mental stage of development.

Chapter 2

Installation of pgs package

2.1 Prerequisites

R is a free statistical software which consists of many packages (see `http://www.R-project.org`). R is available for many platforms (PC/Windows, PC/Linus, Mac/OsX,...). It is assumed here that R is already installed.

We have not tested the pgs package under old versions of R. A version of R $\geq$ 2.7 should fit. The pgs package depends on three other R packages: check that they are already installed or install them if not.

- The package *methods* is a standard package of R. It should be already installed in most cases. If not, it can be downloaded and installed from the Web site of R.
- The package *adapt* performs numerical integration. It can be downloaded and installed from the Web site of R.
- The package *gsl* is a front end to the GNU Scientific Library. It can be downloaded and installed from the Web site of R. Linux users should install first the Gnu Scientific Library (`http://www.gnu.org/software/gsl/`.

2.2 Obtaining and installing pgs

The installation file for pgs package can be downloaded at the URL `http://w3.jouy.inra.fr/unites/miaj/public/perso/KienKieu/pgs`. Note that this package is distributed under the CeCILL free license which is rather similar to the GPL license.

Installation of pgs is standard: follow the general instruction given in R documentation for installing packages.

In order to load pgs package in a R session, just type

```
> library(pgs)
```

in the R command window. The documentation of pgs is available in the on-line help of R.

Chapter 3

Lattice of figures

The sampling devices considered here are lattice of figures. A *figure* is a simple geometric test set such as a single point, a finite point pattern, a quadrat, a segment, a line or a plane. By translations, the figure is repeated in space at regularly spaced locations. The translation vectors form a *vector lattice*.

3.1 Classes of geometric objects

In pgs package, geometric entities are represented by objects. The package operates on three classes of geometric objects:

- Simple figures are represented by objects of the class *Figure*.
- The vector lattices are represented by objects of the class *VecLat*.
- The lattices of figures are represented by objects of the class *FigLat*.

Note that the classes have been designed for lattices and figures in spaces with arbitrary dimensions (not only in 2D and 3D spaces).

Most users will handle only *FigLat* objects generated using predefined generating functions, see Section 3.2.

3.2 Specific generators of some particular planar lattices of figures

The package provides specific generator functions for some common planar lattices of figures:

- The figure can be a finite point pattern, a quadrat, a segment or a line.
- The 2D vector lattices are rectangular hexagonal or quincunx.

For instance, the command line

```
> plat = PPHexLat2(delta = 1, n = 4)
```

creates an hexagonal lattice of point patterns with 4 points lying in a square of side length equal to 0.2. The argument delta determines the distance between homologous points in two neighbour point patterns. The second argument n gives the number of points in each point pattern.

The object plat is printed as follows:

```
> print(plat)
```

```
2D lattice of point patterns
Vector Lattice:
An object of class "VecLat"
dimspace:  2
dimsupp:  2
determinant:  0.8660254
generating matrix:
     [,1]      [,2]
[1,]    1 0.5000000
[2,]    0 0.8660254
determinant:  0.8660254
Figure: PointPattern
coord:
     [,1] [,2] [,3] [,4]
[1,]    0  0.2  0.2  0.0
[2,]    0  0.0  0.2  0.2
```

The function print displays the generating matrix of the hexagonal vector lattice with its determinant and the column matrix coord containing the Cartesian coordinates of the points.

Other arguments can be passed to the function PPHexLat2: hp and vp determine the "size" of the point pattern bounding rectangle and h3 defines the orientation of the point pattern.

Similarly, rectangular and quincunx lattices of point patterns can be created by using the specific generator functions PPRectLat2 and PPQcxLat2.

For planar lattices of quadrats or segments, use the generators QHexLat2, QRectLat2, SHexLat2 and SRectLat2. The function LLat2 creates planar lattices of lines.

For a complete documentation of these functions, see the on-line Help.

A plot function is implemented for planar lattices of figures. The command line

```
> plot(plat, xlim = c(0, 3), ylim = c(0, 3), pch = "+", cex = 0.3)
```

provides the graphic shown in Figure 3.1.

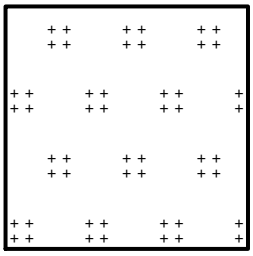

Figure 3.1: Plot of the object plat representing a lattice of point patterns.

3.3 Specific generators of some particular 3D lattices of figures

The package provides three specific generators of 3D lattices of horizontal point patterns: rectangular , body-center rectangular and face-centered rectangular lattices of horizontal point patterns. For instance, the command line

```
> p3lat = PPRectLat3(dx = 1, dy = 1, dz = 1, n = 4)
```

creates a 3D cubic lattice of point patterns with 4 points lying in a horizontal square of side length equal to 0.2.

Body-centered and face-centered lattices of horizontal point patterns can be created by using the generators PPBCRectLat3 and PPFCRectLat3.

Complete documentations of the generators are available in the on-line Help.

3.4 Generator FigLat

The generator function FigLat can be used to create user-defined lattices of figures.

The lattice of figures plat defined above can also be obtained by the command line

```
> plat = FigLat(2, HexLat2(sqrt(3)/2), PP2(n = 4, h = 1/5))
```

- The first argument passed to FigLat is the dimension of the space where the structure lies (2 for planar structures).
- The second argument passed to the function FigLat is a *VecLat* object representing the vector lattice.

Such a vector lattice can be created by the general generator VecLat. The unique argument of the function VecLat is the generating matrix of the vector lattice.

Planar hexagonal lattices may be generated in a simpler way by the specific generator HexLat2. The argument of the function is the determinant of the generating matrix, that is the area of a fundamental tile of the hexagonal lattice. Similarly, the generator RectLat2 is used for planar rectangular lattices.

- The third argument defines the figure, here a point pattern represented by a *PointPattern* object.

  The generator PointPattern generates such objects, it takes as a unique argument a column matrice containing the Cartesian coordinates of the points.

  The specific function PP2 creates some particular planar point patterns. The 3D generator PP3 converts these planar point patterns into horizontal 3D point patterns.

Chapter 4

Planar area prediction

The spatial structure of interest is considered as a random bounded planar set. Note that the case where the structure is deterministic may be considered as a special case of this framework. The feature to be *predicted* is the (total) area of the structure. The term "prediction" is preferred to estimation because the area is a random variable instead of a fixed parameter.

The structure under study is sampled by a lattice of figures, see examples in Figure 4.1. The area predictor is the "content" of the intersection of the structure with the lattice of figures divided by the sampling intensity (mean content of the lattice of figures seen in a region with unit area). For instance, if the sampling device is a lattice of points, the area predictor is equal to the number of lattice points hitting the structure multiplied by the area of a tile of the lattice. If the sampling device is a lattice of quadrats, the area predictor is the area of the structure observed in the quadrats divided by the fraction of the plane covered by the quadrats.

If the lattice of figures is uniformly randomly translated, the area predictor is unbiased: the conditional mean prediction given the structure is equal to the area. The approximations for the prediction MSE as given in [22] are obtained under the assumption that the normals to the boundary of the structure are isotropically distributed. Note that if the lattice of figures is isotropically randomly rotated, the MSE approximations hold even for anisotropic boundaries.

4.1 MSE approximation formulae in the isotropic case

Under the assumption of isotropy, the MSE approximation formulae involve two terms:

- one term depending only on sampling parameters and involving the multidimensional Epstein Zeta function.

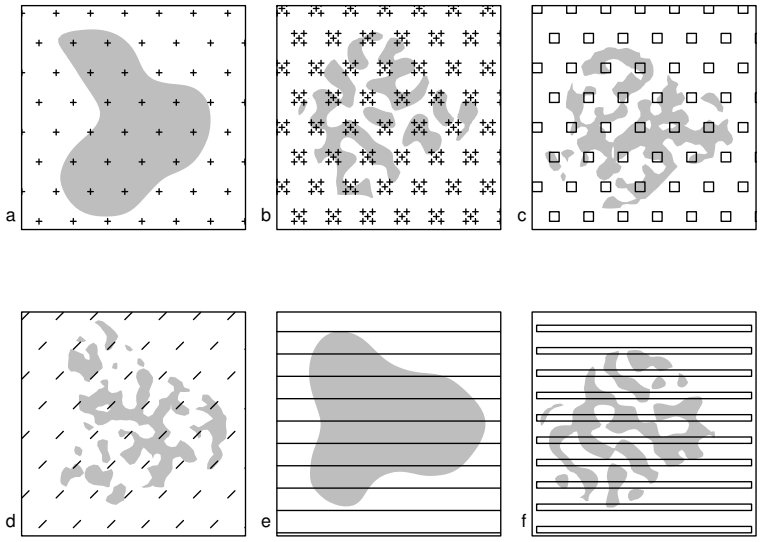

Figure 4.1: Examples of lattices of figures used for area prediction. A lattice of figures is superimposed on a planar structure (shown in grey). The total area of the planar structure is predicted from point counts, length or area measurements. (a) Hexagonal lattice of points, (b) lattice of point patterns, (c) lattice of quadrats, (d) lattice of segments, (e) lattice of (infinite) lines, (f) lattice of (infinite) strips.

- one term depending on the boundary of the structure under study: the mean perimeter.

The MSE approximation is computed using the function area.mse:

```
> area.mse(plat, B = 1, L = 3)

[1] 0.02222672
```

- The first argument plat is the sampling test system.
- The second argument B is the mean perimeter of the structure (to be provided).
- The third argument is an integer used as a criterion for stopping summation of the Epstein Zeta function.

4.2 Practical design of a sampling scheme

Let us consider the case where the figure is fixed and the vector lattice is defined up to a scaling parameter. The function latscale computes the scaling parameter u such that the prediction coefficient of error is equal to a given value.

```
> u = latscale(PPHexLat2(n = 4), A = 1, shape = 5, CE.n = 0.05,
+     upper = 2)
```

- The first argument passed to the function is the lattice of figures. The associated vector lattice is supposed to be unit (i.e. its determinant must be equal to 1). The lattice of figures plat created in Section 3.2 does not fulfill this condition (determinant equal to 0.8660254. The vector lattice associated with the lattice of figures defined by PPHexLat2(n=4) is unit.
- The second argument A is a rough estimate of the mean area.
- The third argument is a rough estimate of the shape parameter $B/\sqrt{A}$. It may be determined approximately using the nomogram provided in [12].
- The fourth parameter is the nominal coefficient of error.
- The fifth parameter is an upper bound for the interval where u is to be searched.

Hence in order to get area predictions with a coefficient of error equal to 5%, one should use an hexagonal lattice where the spacing between homologous points is equal to 0.3487114.

4.3 A posteriori MSE computation

Let us consider the case where a series of say 10 structures have been sampled. The sampling lattice of figures is represented by plat, see Section 3.2. The point counts are given by

```
> counts = c(5, 3, 1, 4, 1, 3, 4, 2, 4, 5)
```

The area predictions are computed as follows

```
> areaPred = counts/(4 * plat@vlat@det)
> areaPred

 [1] 1.4433757 0.8660254 0.2886751 1.1547005 0.2886751 0.8660254 1.1547005
 [8] 0.5773503 1.1547005 1.4433757
```

The prediction variance is

```
> var(areaPred)

[1] 0.1814815
```

The prediction variance is the sum of the area variance and of the area prediction MSE. The MSE formula as given in [22] requires parameters depending on the lattice of figures and the mean perimeter of the structure under study. Let us consider the case where the shape parameter $B/\sqrt{A}$ may be approximated by 5 in view of the nomogram given in [12]. The mean perimeter is approximated by

```
> B = 5 * sqrt(mean(areaPred))
> B

[1] 4.805623
```

The MSE is computed using the function area.mse:

```
> area.mse(plat, B)

[1] 0.1068132
```

The area variance may be computed as the difference between the prediction variance and the prediction MSE.

4.4 MSE estimation

4.4.1 MSE estimation in the isotropic case

If an estimate of the mean perimeter is available it can be provided to the function area.mse in order to compute a MSE estimate. For instance, this is the case when the sampling scheme is designed for estimating simultaneously the mean area and the mean perimeter of the structure.

An alternative approach consists in using only data collected for area prediction. It is supposed that individual data for each sampling figure are available. The mean perimeter estimation is based on short-range data covariations (empirical covariogram near the origin) and depends on the sampling figure geometry. Then the MSE can be estimated by providing that mean perimeter estimate to the function area.mse.

All estimates computations can be done with the function area.mse.est.

As an example, consider the case where a planar structure is sampled by the rectangular lattice of point patterns pplat defined by

```
> pplat = PPRectLat2(1, 1, 5, 0.4)
```

The first and second arguments are the horizontal and vertical spacings of the vector lattice. The third and fourth arguments define the point pattern : 5 points lying in a square of side length equal to 0.4 (4 points lying at the square corners and one middle point).

The data to be used are contained in the matrix pp.counts displayed as

```
0 0 0 0 0 0 0 0 5 5 0
0 0 4 5 3 0 0 0 4 5 1
0 0 0 0 0 0 0 2 5 5 4
0 0 0 0 0 0 5 5 5 5 5
1 0 0 0 0 5 5 5 0 2 5
5 0 0 0 1 5 3 0 0 0 0
3 1 0 1 5 3 0 0 0 0 0
0 4 4 5 5 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
```

Each entry of the matrix pp.counts contains the individual data of a sampling point pattern.

The sampling grid parameters and the collected data are passed as arguments to the function FigLatData.

```
> ppldata = FigLatData(pplat, pp.counts)
```

Both the mean perimeter and MSE estimates can be computed with the function area.mse.est:

```
> area.mse.est(ppldata, mse.only = FALSE, iso = TRUE)
```

```
$B.est
[1] 41.47922

$mse.est
[1] 0.5150249
```

- The first argument passed to the function is the lattice of figures and the array containing the data. The data array is reduced to a matrix (as in the example) when sampling is not repeated.
- The second argument is set equal to TRUE (the default value is FALSE) to specify that estimates must be computed under the assumption of isotropy.
- The third argument is set equal to TRUE (default value) if both the mean perimeter and MSE estimates are required. Otherwise, only the MSE estimate is provided.

The function area.mse.est has an extra argument diff2use defining the data covariations to be used for the MSE estimation. See the on-line Help for more details.

4.4.2 MSE estimation under mild anisotropy

The MSE estimation method can be extended to the case where the boundary can be made isotropic by a specific area preserving deformation: stretching the structure along a given direction, compressing it in the orthogonal direction. For sake of simplicity, this transformation is called stretching. The stretching is applied both to the structure and the lattice of figures. For the extension to be valid, the data provided to the function area.mse.est must be invariant under the stretching. This is the case, for example, for lattices of point patterns and quadrats.

The function area.mse.est first estimates the stretching in order to provide the mean perimeter and the MSE estimates.

For the lattice of point patterns and the data given in pppldata, one gets

```
> area.mse.est(ppldata, mse.only = FALSE, iso = FALSE)

$B.est
[1] 40.47671

$deformation
           [,1]       [,2]
[1,] -0.9363625 -0.6710058
[2,]  0.5056472 -0.7056111
```

```
$mse.est
[1] 0.468871
```

The argument iso is set equal to FALSE to specify that a stretching has to be applied.

The component list deformation returned by area.mse.est defines the stretching matrix.

Bibliography

[1] L. M. Cruz-Orive. Estimating volumes from systematic hyperplane sections. *Journal of Applied Probability*, 22:518–530, 1985.

[2] L. M. Cruz-Orive. On the precision of systematic sampling : a review of Matheron's transitive methods. *Journal of Microscopy*, 153(3):315–333, 1989.

[3] L. M. Cruz-Orive. Systematic sampling in stereology. In *Proceedings of the 49th Session of the International Statistical Institute*, volume 2, pages 451–468, 1993.

[4] L. M. Cruz-Orive. Precision of Cavalieri sections and slices with local errors. *Journal of Microscopy*, 193:182–198, 1999.

[5] L. M. Cruz-Orive and Myking. Stereological estimation of volume ratios by systematic sections. *Journal of Microscopy*, 122:143–157, 1981.

[6] M. Garcia-Fiñana and L. M. Cruz-Orive. Explanation of apparent paradoxes in Cavalieri sampling. *Acta Stereologica*, 17:297–302, 1998.

[7] M. Garcia-Fiñana and L. M. Cruz-Orive. Fractional trend of the variance in cavalieri sampling. *Acta Stereologica*, 19:71–79, 2000.

[8] M. Garcia-Fiñana and L. M. Cruz-Orive. New approximations for the efficiency of cavalieri sampling. *Journal of Microscopy*, 199:224–238, 2000.

[9] X. Gual Arnau and L. M. Cruz-Orive. Consistency in systematic sampling. *Advances in Applied Probability*, 28:982–992, 1996.

[10] X. Gual Arnau and L. M. Cruz-Orive. Variance prediction under systematic sampling with geometric probes. *Advances in Applied Probability*, 28:982–992, 1998.

[11] X. Gual Arnau and L. M. Cruz-Orive. Systematic sampling on the circle and on the sphere. Technical report, Departamento de matematicas, estadistica y computacion. Universidad de Cantabria, 1999.

[12] H. J. G. Gundersen and E. B. Jensen. The efficiency of systematic sampling in stereology and its prediction. *Journal of Microscopy*, 147(3):229–263, 1987.

[13] H. J. G. Gundersen, E. B. V. Jensen, K. Kiêu, and J. Nielsen. The efficiency of systematic sampling in stereology - reconsidered. *Journal of Microscopy*, 193(3):199–211, 1999.

[14] A. M. Kellerer. Exact formulae for the precision of systematic sampling. *Journal of Microscopy*, 153(3):285–300, 1989.

[15] D. G. Kendall. On the number of lattice points inside a random oval. *Quarterly Journal of Mathematics. Oxford. Second Series*, 19:1–26, 1948.

[16] D. G. Kendall and R. Rankin. On the number of points of a given lattice in a random hypershere. *Quarterly Journal of Mathematics. Oxford. Second Series*, 4:178–189, 1953.

[17] K. Kiêu and M. Mora. Asymptotics for geometric spectral densities and a stochastic approach of the lattice-point problem. *Mathematicae Notae*, 2004.

[18] B. Matérn. Estimating area by dot counts. In J. Lanke and G. Lindgren, editors, *Contributions to Probability and Statistics in Honour of Gunnar Blom*, pages 243–257. Department of Mathematical Statistics, University of Lund, 1985.

[19] B. Matérn. Precision of area estimation: a numerical study. *Journal of Microscopy*, 153(3):269–284, 1989.

[20] G. Matheron. *Les variables régionalisées et leur estimation.* Masson, 1965.

[21] G. Matheron. The theory of regionalized variables and its applications. Technical report, Centre de morphologie mathématique, École des Mines de Paris, 1971.

[22] M. Mora and K. Kiêu. Precision of stereological planar area predictors. *Journal of Microscopy*, 222(3):201–211, 2006.